#2

Information for IDS

NT0091US

1. Prior Art described in the specification

1-1. Article

(1) THE SMART CARD Handbook, 1997, John Wiley & Sons, pp.261-267

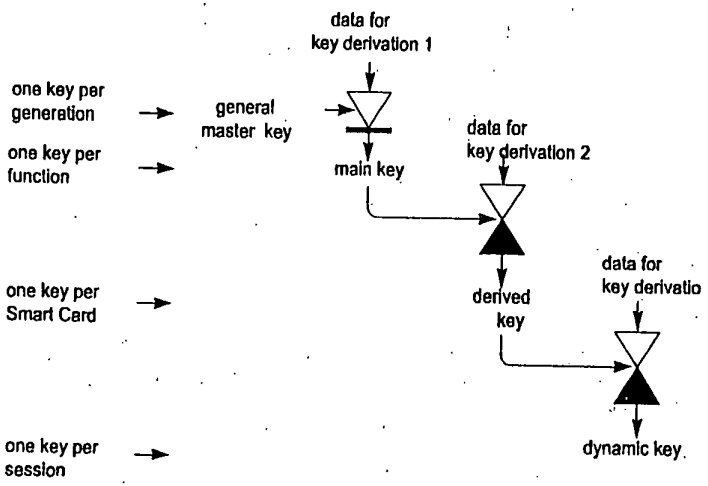

Figure 8.17 Example of Smart Card key hierarchy

The next step in the key hierarchy is represented by the derived keys. These are the keys located within the cards. Each card contains a set of derived keys, subdivided by generation and function. When such a card is used at a terminal, the latter can calculate the relevant derived keys from the derivation data. Naturally, the key derivation data are first read by the terminal from the card. The dynamic key, unique to the session and only valid for its duration, can now be calculated in the next step with the derived key. In typical Smart Card applications, the dynamic key would be used for between a minimum of a few hundred milliseconds and a maximum of a few seconds, and then scrapped.

The system just described may appear at first sight to be expensive and complicated, but in comparison with real systems this is not the case. The purpose of such a system is to specify a precise method for generating all of the keys. It also indicates implicitly what measures must be taken if a key should ever become known. If this is the general master key, a new generation must be introduced in order to allow the system to continue operating without security restrictions. In contrast, if a master key becomes known then only the branch below it needs to be barred or switched to a new generation. If it is a derived key which is no longer secret, then the only step which needs to be taken is to block the relevant card. Any additional key management measures would be quite superfluous. All these steps assume, of course, that the reason for the key being revealed could be discovered and could be avoided in future.

These key hierarchies naturally require a great many keys to be generated and stored in Smart Cards. However, several functions can be combined in the same key to save storage space. It is also quite possible to design the hierarchy in a different way. This depends to a large extent on the actual system for which the key manager was developed.

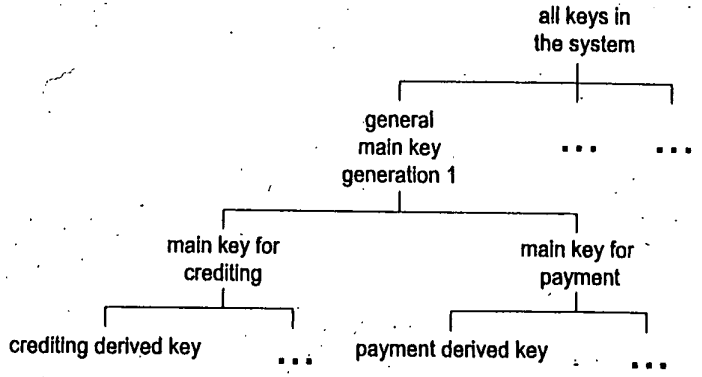

Figure 8.18 Example shows key in electronic purse with the functions of loading and payment. (Only stored keys are shown, i.e. dynamically created keys have been omitted in order to simplify matters)

8.5 SMART CARD SECURITY

The following four components guarantee the security of a Smart Card. The first security component is the body of the card, into which the microprocessor is incorporated. Many of the security features exploited here are not only machine-readable, but can additionally be checked visually. The technology is not specific to Smart Cards, and is used elsewhere in the card industry. The other components – semiconductor, system software and application – protect the data in the card's microprocessor.

It is only when all these components are present and their defence mechanisms are operational that a Smart Card's security is guaranteed. Where the card only finds its application in the machine-readable domain, the card-body component is unnecessary. The three other components, those which are distinct from the body, are vital, however, for the logical and electronic safety of the card in order to protect it from tampering. If even one of these three components fails or does not meet its specifications, the card is no longer secure. These components are, therefore, linked by a logical AND function.

8.5.1 Technical options for chip hardware

The technical options for protecting a Smart Card microprocessor against tampering are manifold. These options should be fully exploited. The special hardware solutions for microprocessors arose from technical security requirements. The Smart Card industry no longer uses even one single standard module; only specialized modules are employed.

Hardware protection may be divided into passive and active elements. The former are based directly on the semiconductor manufacturing technology. They cover all the

possibilities and procedures which can be utilized to protect the memory and the rest of the microprocessor's functional elements from various types of analysis.

The semiconductor (passive) solutions are complemented by a whole spectrum of active chip components. Active protection means the additional integration of various sensors on the silicon crystal. These sensors are interrogated and evaluated by the card's software as and when necessary. Of course, this is only possible if the chip is connected to all of its power supply lines and these are fully activated. A chip which does not carry an electric current cannot sense any signals, let alone evaluate them. That is why for sensors, more than any other component, the gap between necessary defence elements and technical gamesmanship is often quite small. A light sensor on the chip which is designed to prevent optical analysis of the memory will not react when the chip is placed on a microscope-carrier if its power and clock supply lines have been disconnected. Such a sensor would also be easy to spot, and could well be obscured by a drop of black ink so that its protective function is neutralized without much difficulty even under operational conditions.

At the end of the day, only long-term functional security is meaningful. For example, a temperature sensor which detects transient and trivial overheating and causes the card's software to clear the entire EEPROM on security grounds does not contribute to increased functional security or protection against attack. For this reason, only a few sensors are implemented in the card as a rule.

8.5.1.1 Passive protective mechanisms

The various options available for passive mechanisms in Smart Card microprocessors are extremely diverse. The most important ones, and those most commonly used in practice, are explained below.

All microprocessors possess a so-called test mode, which is used to test the chip during manufacture and to perform internal test routines. These are performed either on the semiconductor while it is still part of the wafer, or in the module on the manufacturer's premises. However, this mode allows access to memory in ways which are strictly prohibited at a later stage. For technical reasons relating to production methods, it is absolutely necessary to be able to read data from EEPROM whilst in this mode.

Conversion from test mode to user mode must be irreversible. This can be achieved by adding polysilicon fuses to the chip. A voltage is then applied to a predetermined test point on the silicon crystal which causes the relevant fuse to burn through. The chip is now converted to hardware user mode. Generally speaking, this process is irreversible, but since it is at least theoretically feasible to bridge the relatively large and flat structures of a burnt fuse on the chip, additional protection is provided in an EEPROM area which cannot be erased. If this contains a particular unalterable value, then the chip switches irreversibly into user mode.

Chip structures (width of conductor tracks, transistor size, etc.) are at the limit of the technically feasible. Conventional widths range from 0.8 µm to 1.2 µm, which in itself no longer presents a particular technical challenge. Transistor density on the semiconductor is about as high as is currently possible using standard lithographic manufacturing processes. It is only these high-precision structures which make it almost impossible to obtain data from the chip through analytical processes.

The chip's internal buses which link the processor to the three memory types – ROM, EEPROM and RAM – are not connected to the outside world, and cannot be accessed, even using very complex methods. An interloper is therefore unable to tap into the microprocessor's address, data or control bus, or to tamper with them and thus read out data from memory.

Many Smart Card microprocessors scramble the internal buses used to control memory. This means that the individual conducting tracks do not lie next to each other in an ascending or descending sequence, but are swapped around and interchanged several times. This is an additional hurdle for the potential hacker to overcome, since at first, he will be unaware of which bus performs which function. But since this scrambling of the tracks is absolutely static, i.e. it is identical in every chip, it constitutes no huge problem in the medium term to identify the buses and to take this into account when tapping in.

The processor's design is also a security factor. It must draw an approximately equal amount of power when performing any machine instruction. If this is not the case, then the current consumed when the most recent instruction is executed may reveal confidential information.

Most programs are stored in ROM. The contents of ROMs commonly used in industry can be read bit-by-bit with an optical microscope. Combining these bits into bytes, and the bytes into a complete ROM code, is not a real problem. It is precisely in order to counter such analysis that the ROM is not located in the top silicon layers, the ones most susceptible to inspection, but in the lower ones. This prevents optical analysis. Nevertheless, sanding down the chip from the back would make it possible to read the contents of the ROM.

Another risk lies in the analysis of electric potentials in the chip while it is in operation. With sufficiently high readout frequency, it is possible to measure charge potentials, i.e. voltages, across very small crystal areas and thus obtain information about the data in the RAM during operation. This can be very reliably avoided by additional metallization over the relevant memory cells. If these metallic layers are removed chemically then the chip will no longer be functional, since it needs to use them as electrical conductors for its operation.

8.5.1.2 Active protective mechanisms

After microprocessor manufacture of the silicon chip, a passivizing layer is added to prevent oxidization or other chemical processes from attacking the chip's surface. The first step in any chip manipulation requires the removal of this layer. A sensory circuit can determine whether it is still present by making resistor or capacitor measurements. If it is absent or damaged, an interrupt can be triggered via the chip software or else the whole chip can be disconnected from the hardware, thus completely preventing any dynamic analysis from taking place.

Every Smart Card microprocessor contains a voltage regulator. It is responsible for a defined disconnection of the module when the upper or lower limits of operational voltage are exceeded in either direction. This guarantees to the software that operation is impossible in those regions in which the chip is not fully functional. If such a regulator were not installed, then in those areas the processor's program counter, for instance, might no longer operate in a stable manner, which would cause uncontrolled jumps within the program.

The power-on detector is a sensor present in all chips which is partially reliant on the voltage regulator. It is independent of the reset signal, and ensures that the chip always works in a defined domain when booted. The reasons for this are the same as for the voltage regulator.

The Smart Card's clock is always supplied from outside, so that computational speed is determined entirely externally. This theoretically permits the microprocessor to be run from outside in single step mode, which provides an excellent opportunity for analysis, particularly in the measurement of power consumption and electric potentials in the chip. A functional low frequency detection module is integrated into the chip, in order to combat tampering of this kind by disconnecting unauthorized applied clocks with such frequencies. Most chip specifications allow clock frequencies down to 1 MHz. Since detection of low chip frequencies necessarily introduces a fair amount of measurement scatter, the chip is usually switched off only if the frequency drops to below 600 kHz. This ensures that the chip can always be operated at the minimum value of 1 MHz. Sometimes high frequency detection is also implemented but modern hardware is so constructed that the chip cannot be operated if the frequency is too high.

Processor features can also offer increased security in Smart Card operation. They consist of additional functional units which can be tested by the terminal in parallel with chip software. Both analogue and digital modules are used. The security of these features is based on camouflage and varies with the application, resulting in application-specific chips.

One sensor variant which is used for temperature supervision is implemented in some chip types but its purpose is controversial. A transient increase in temperature beyond the specifications does not cause permanent damage to the chip nor does it necessarily mean that an attempt at tampering is taking place. Switching the chip off in such borderline situations could lead to artificially increased failure rates, rather than offering any additional security to the suppliers of Smart Card systems.

8.5.2 Software protection mechanisms

The mechanisms which protect system software must build on those used for hardware. No gap in this ringfencing must be overlooked, since the protection mechanism's three components (hardware, operating system, application) are linked through a logical AND function. If one mechanism fails, the whole security setup falls apart. The operating system itself supports the actual application, whose data and routines must be protected.

During operating system initialization, the most important sections and those of the hardware need to be tested. A RAM test is imperative, since it stores all access conditions during run-time and the failure of a particular bit may cause the whole security system to crash. The calculation and comparison of checksums over the EEPROM's most important sections is equally vital.

Completion of the operating system, i.e. the loading of tables, program code and configuration data into the EEPROM, represents not merely increased flexibility but is also an important security factor. The result is that the chip manufacturer, who receives the complete and assembled ROM program code for mask fabrication, has incomplete knowledge of the operating system. Those system parts which are located in EEPROM remain unknown to him, so that analysis of the ROM code does not reveal everything there is to know about security mechanisms and system functions.

Layer separation, with clearly defined interfacing parameters between the individual layers, is always a feature of stable and robust Smart Card operating systems. The adverse effects resulting from programming or design faults which may exist within the operating system are minimized by clean separation of the layers within it. Although this does not mean that errors cannot occur, their repercussions are not as serious as in a system programmed by using very compressed and concise code, since the separation means that errors cannot easily be propagated from one layer to the next.

Another highly important security element, namely I/O control, protects the memory from unauthorized access. The entire communication to and from the card takes place across the I/O interface controlled by the operating system. No other access is possible. This represents the strongest memory protection in the card, since this is the only way in which it is possible to ensure that the system retains control over memory access in all situations.

The transmission protocol controlled by the transport manager must intercept all possible false inputs. There must be no way of influencing data transmission by means of manipulation of the transmission blocks resulting in false data being sent without authorization from the memory to the terminal.

File organization, and in particular the headers (i.e. the file descriptors), should be secured via checksums. If some data is altered inadvertently, at least this must be detectable by the operating system. In view of the fact that the respective object-oriented access conditions are located in this part of the file, this requirement is evidently very important.

Some systems encapsulate and insulate the individual DFs containing the applications with their files from each other. These concepts are based only on pure software protection and are unsupported in the chip hardware. This makes protection less effective than it might otherwise be. However, in the event of a fault this is of great benefit, since it makes it impossible for the file manager to exceed the boundaries of a DF without prior explicit selection. Hence the results of a memory error in a file are, at least, restricted to that file.

All the memory sections in the EEPROM which are of vital importance for the card's operating system, must be protected by checksums (EDC). When these sections are accessed or interrogated, the consistency of their contents must first be established through the checksums, so that the system's stability is not endangered by EEPROM memory errors.

The writing of data to the EEPROM requires the chip's charge pump to be switched on. This increases the card's power consumption considerably, and this can be measured using a simple device. This means that system design must take into account that writing to the EEPROM can always be detected outside the card. The software must prevent an interloper from making use of this knowledge. It is therefore very important for current measurement not to allow any useful information to be deduced about internal routines and decisions made within the machine program. For example, if a current measurement made it possible to make a reliable deduction about PIN comparison before the completion of an instruction and receipt of the return code, this could be used to good effect to analyse the PIN.

Early Smart Card applications were always based on a centrally administered access mechanism. These centralized access automata suffered from the disadvantage that software or memory errors affected the entire security of the card. Modern, object-oriented file management systems with access rights localized to individual files have the advantage that in the event of memory errors only one file is affected and the other files' security remains intact. This is a fundamental property of distributed systems. Their programming is somewhat more costly, but they have considerably more resistance to tampering or failure, thanks to the security offered by this local authorization.

The operating system must be capable of completely deactivating the card. This is very important during the last phase of its life. Statistical methods, together with the collection of expired but fully functional cards, make it possible to perform very accurate analyses of the chip's software. In order to prevent this, the operating system must have mechanisms at its disposal which enable it to deactivate itself completely together with all its programs, and thus render electrical or run-time analyses impossible.

8.5.3 The application's protective mechanisms

The application's protective mechanisms are based on those which are present in the hardware and the operating system. The application has to rely on the two lower layers performing their protective tasks with the utmost rigour, since it cannot intercept a hardware or system error. For example, if the EEPROM can be read by an analytical procedure, then the most complex and secure encryption methods become useless, since the keys can be obtained by the attacker directly from the EEPROM. However, an application must be so constructed that if it compromises the card, the whole system is not compromised as a result.

It is of great importance to have a unique card number which is not duplicated in any other card. This allows the card to be identified unambiguously across the system. In addition, this number can be used for key derivation and offers the possibility of setting up warning lists so that suspect cards can be removed from circulation.

A hacker's task is made considerably harder if the files on the card are protected not only by the access conditions attached to the object, but also by instructions and corresponding parameters being fixed and predetermined by a state automaton. This stops trial-and-error experimentation with instructions, or combinations of instructions, from uncovering system-specific characteristics which may eventually be exploited. If the instruction sequences are supervised by a state automaton, only the instructions defined in the application can be performed, all the others being blocked by the state automata before they can be implemented. This largely reduces the options open to a hacker through the manipulation of instructions.

Data transmission in potentially insecure environments contains several risks. It is possible, through relatively straightforward technical manipulation of the card/terminal interface, to add almost any desired data during a session or delete it from the normal run. If this happens during the transmission of sensitive data, an attacker may, under certain circumstances, obtain some benefit from it. The secure messaging procedure can be implemented so as to interfere with such tampering which is neither excessively expensive nor difficult. Complete data encryption should nevertheless be avoided as far as possible, and should only be used to transmit such data as secret keys. The reason for this is the double encryption required for the transmitted data, which reduces effective transmission speed rather heavily. Another argument against complete encryption derives from data protection legislation. Almost all the data written to the card's memory is in the public domain. If it is encrypted, then nobody can check what was actually written to the card or read from it. In order to counter any legitimate objections, the data should not be encrypted during transmission if possible.

If a session is interrupted by an undefined breakdown, or the details of a previous session are fundamentally unclear, it is very useful to have application-specific protocol files

available in the card. During the session, these are updated by the operating system using the current application states and any signatures or other terminal data received. This data is then stored in a cyclical file, the oldest record always being overwritten and thus lost. If this file contains, say, 20 records, then the data concerning the most recent 20 sessions can be stored for analysing the session. This allows many ambiguities to be removed, and disputed transactions and events clarified. Another argument for maintaining detailed transaction records is that error recovery functions are then made possible. This would allow the card's previous state to be automatically restored after an undefined breakdown, without the need for human involvement in the analysis of the precise run and sequence of events during which this had occurred.

At the end of the day, one-sided authentication (as practised on magnetic cards) means that only the terminal can be used to check whether the card is genuine. Due to its passive nature, this type of card cannot check to see whether the terminal itself is authentic. The introduction of Smart Cards has changed this situation fundamentally, and the card is also capable of checking to see whether it has been inserted into a genuine terminal. This has far-reaching consequences in security terms, since it renders the card actively able to take steps against unauthorized access. The options made available by this ability to carry out mutual authorization are enormous, and are nowhere near exhausted. At the very least, a Smart Card should be able to lock itself against further unauthorized access, in whatever form, as long as the terminal is unable to authenticate itself correctly. This makes it impossible to undertake backroom analysis of Smart Card operating systems, even if this is only done so as to examine all the available instructions.

Terminals with security modules can run Smart Card applications wholly independently. Naturally, periodic uploads and downloads to and from a background system are necessary, but in the normal course of events this only happens rarely. In large applications with a large number of cards in circulation, the terminals must be capable, if necessary, of establishing communication with the background system without delay, so that the latter can communicate directly end-to-end with the card. This becomes increasingly important with the size of the system, since the larger the system the more benefits are available to a hacker. With direct connection between the Smart Card and the background system, the latter can access its current database and lock the card if appropriate. Furthermore, the keys are much more securely stored in a background system than in the many terminals out in the field, even if they are fitted with a security module. In addition, these sporadic end-to-end communications allow the background system to make very efficient statistical analyses about diverse data contained in the cards. Of course, all these arguments are mainly of interest to the electronic wallet application. The on-line status can be forced on the card by random variables or by time limits stored within it. Counters in the card which can demand an on-line connection with mutual authentication when they reach a particular value are just as effective. The background system can reset the counter once connection is established.

8.6 TYPICAL ATTACK AND DEFENCE MECHANISMS

This section presents and explains several examples of types of interference which have almost become classics in their field. Protective measures against these attempts are also described. They, in turn, can be circumvented by slightly modified attack scenarios